CS635: Capturing and Rendering Real-World Scenes

Instructor: Daniel G. Aliaga
Classroom: TBD
Time: MWF @ 10:30-11:20am
Office hours: arrange by email

1. Course Overview

The objective of this course/seminar is to understand the fundamental problems and challenges encountered when capturing, modeling, and rendering (and printing) 3D structures and objects. The course covers several subjects within computer graphics, computer vision, artificial intelligence, and computer science so as to provide to the student a full understanding of the capture/model/render pipeline. From this understanding and cross-fertilization of ideas, it is expected that students will in the future be able to develop new and improved approaches.

2. Tentative Schedule

Week of	Lecture	Assignment	Final Project
Jan 13	Introduction, Toolbox (e.g., features, deblurring. optimization: minimization, least squares, simulated annealing, MCMC, machine learning and AI, human computation); Camera calibration	**Jan 17: #0 out**	
Jan 20	Camera Calibration	**#1 out**	
Jan 27	Passive Acquisition (e.g., stereo, linear, bundle adjustment, pose-free formulation)		
Jan 30	Active Acquisition (e.g., lasers, structured light)	**#2 out**	
Feb 3	Photogeometric Stereo (e.g., photometric stereo, photogeometric method)		
Feb 10	Deep Learning Enhanced Reconstruction	**#3 out**	
Feb 17	Single-Image (DL) Reconstruction		
Feb 24	Forward/Inverse Light Transport	**#3 in**	Suggestions given
Mar 3	**Project Background Presentations**		Declare projects
			Background presentations
Mar 10	(Deep) Computational Images & Cameras		
Mar 17	Spring break (no classes)		
Mar 31	(Deep) Displays and Inverse Optics		
Apr 7	**Mid-Project Presentations**		Mid-project presentations
Apr 14	TBA		
Apr 21	3D Printing and Design		
Apr 28	**Demo week**		**Demo and presentation!**
	(details forthcoming…)		

For more information about the general type of research, I recommend looking at:

CGVLab Webpage: http://www.cs.purdue.edu/cgvlab

My webpage: http://www.cs.purdue.edu/~aliaga

For graphics in general: http://kesen.realtimerendering.com/

For vision in general: https://openaccess.thecvf.com/menu

3. Workload

The course is divided into two parts.

- The first part describes, during the lectures, research methods to be presented by reviewing the latest works in the field. The students will also present informal presentations and summaries about work relevant to their projects. An exact schedule is to be determined once the semester starts.
- The second part of the course consists of a short set of assignments and then a substantial final project. The assignments provide guided programming projects that progressively implement a basic system to build 3D models from images. An initial software package is given so that the students can immediately focus on the algorithms. The core effort of the course is in the final project. There are multiple deliverables for the final project. Further, a publication/submission of the semester project would be an ideal goal.

The course grade is determined by the performance in the programming assignments, the final project, and class participation. Each assignment will be evaluated during an interactive session with the instructor. The grade depends on a combination of meeting the requirements, the presentation, and the sophistication of the solution. There will be no final exam but rather a public demo day at the end of the semester with all projects.

Assignment #0 – Compiling Warm-up (1 week)
Download, compile, and execute the provided software package. The deliverable includes a simple image sequence that is trivial to do with the provided software. The objective is just to "get you up and running for the assignments/project". If you wish to use your our own framework, please see the instructor. The deliverable is a simple video.

Assignment #1 – Camera Calibration (2 weeks)
Capture images (using your camera or a loaned camera) and "calibrate" the camera. The resulting calibrated camera should be used to verify correct pose estimation of a pair of images via simple visual feedback (correspondence can be established, for example, manually via mouse clicking).

Assignment #2 – Example Real-world 3D Reconstruction (2 weeks)
Using the previous assignment, reconstruct a 3D object (using triangles) and render the object within an OpenGL program where you can intuitively control the viewpoint and/or object position and orientation.

Assignment #3 – Deep Learning Based Recognition (2 weeks)
Using the previous assignment and a provided framework, train a neural network to "learn" how to do perform a part of the 3D reconstruction task. Learning will be done on your PC or on a department provided GPU cluster.

Final Project (final due date is end of semester demo)

Feb 24: project ideas will be given to all students in written form
March 3/5/7: in-class presentation of project-relevant papers (presentation length is "long")
Apr 7/9/11: in-class Powerpoint presentation of mid-project progress
Week of Apr 28: projects due (Public Demo)

The grade distribution is tentatively:
Assignments: 25% (1%, 8%, 8%, 8%)
Final Project: 65% (10%, 15%, 40%)
Class Participation and Attendance: 10%

A subset of relevant conferences that could be targeted with this semester's work include:

- **SIGAsia: ACM SIGGRAPH Asia (late May submission deadline)**
- **EG: Eurographics (September submission deadline)**
- **NeurIPS: (May deadline)**
- **ICCV: (March deadline)**

4. Administrative Issues

All assignments must be handed-in by the specified due date/time. An assignment late by up to one day receives a 50% penalty (e.g., if maximum score is 10, it will be a maximum of 5), by up to two days a 75% penalty and after that a 100% penalty. THERE IS NO LATE PASS. The final project consists of 3 formal presentations (initial background research – papers review, a mid-project presentation, and final project presentation). The exact dates will be established once students and projects are settled. All final project related presentations must be on time; otherwise a grade of 0 is given for that component.

Grading is done by submitting system via Brightspace and then demo'ing the assignment, and reviewing code, with individual arranged meetings with the instructor.

All assignments, presentations, and projects must be done individually unless otherwise indicated by instructor. Final projects may be teams of two people (rarely three or more people). In research, it is highly encouraged to "build upon the shoulders" of others, however due credit must be given to the sources. Unreported copying or plagiarism will give you a failing grade in the course and you will be subject to standard departmental and University policies. For the programming assignments, code obtained from the Internet, books, or other sources may *not* be used. For the final project, previously-written code is permissible pending instructor approval.